SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I2P104

Volume 12 Issue 2, 27-38, February 2025

Original Article

Integrated Distance Based Convolutional Neural Network Optimized Using Elephas-Kiboko Algorithm for Movie Recommendation

Sneha Bohra[1], Amit Gaikwad[2]

[1,2]*Department of Computer Science and Engineering, G.H. Raisoni University, Amravati, Maharashtra, India.*

[1]*Corresponding Author : snehab30@gmail.com*

Received: 05 December 2024 Revised: 04 January 2025 Accepted: 02 February 2025 Published: 22 February 2025

Abstract - *A movie recommendation system aims to suggest movies based on users' preferences, past activities, and browsing history. These systems depend on algorithms that process extensive user behaviour data, movie attributes, and contextual information to provide personalized recommendations. This system suffers from several challenges like cold-start, sparse data, or dynamically changing user preferences. To overcome these issues, this research introduces the Elephas-Kiboko Optimization (EKO) Algorithm for the optimization of CNN based movie recommendation model. The proposed model also introduces an integrated distance metric that combines two standard distance metrics, namely Bhattacharya and Euclidean distances, to analyze movie similarities, considering both textual and visual features for enhancing the recommendation accuracy. The Elephas-Kiboko Optimization (EKO) method is a bio-inspired algorithm that utilizes clan-based updates and adaptive behaviors to enable effective navigation through the exploration and exploitation phases. This innovative approach addresses key challenges of recommendation systems and improves recommendation accuracy and user satisfaction. Experimental results on popular movie datasets show that the EK-CNN model delivers outstanding performance for metrics like accuracy, F1-score, precision, and recall with values of 94.87%, 94.65%, 94.00%, and 95.75%, respectively.*

Keywords - *Bio-inspired, Bhattacharya distance, Convolutional Neural Network, Elephas-Kiboko Optimization, Euclidean distance, Movie recommendation system.*

1. Introduction

Over the last decade, the upswing in the use of the internet has significantly increased the volume of available information. To address the issue of information overload, an information filtering algorithm called the Recommendation System (RS) is a primary requisite. These systems instinctively suggest desired content to the user [1-4]. Nowadays, Movie Recommendation Systems have gained popularity since they assist users in selecting their favourite movie from an expansive movie archive. To serve the user with preferred content, considerable efforts have been made to develop innovative algorithms for Movie Recommendations [5].

Although recommendation systems can be developed by using either a Content-Based (CB) or Collaborative Filtering (CF) approach, CF has been notably advanced in recent years. It serves as the backbone for most applications today. CF uses ratings provided by the users to recommend films based on the preferences of similar users. Major online platforms, such as Netflix, YouTube, etc., have implemented a CF-based approach to offer tailored recommendations [6-8, 14].

Although notable improvements have been made with implementing Collaborative Filtering (CF) and model-based approaches, the core issues of Sparse Data, scalability, and the cold-start problem persist. While these systems demonstrate the delivery of potential personalized recommendations, they struggle with adapting to changing user preferences and managing dynamic content updates effectively [15-17].

The model-based algorithm implements dimensionality reduction methods to address the data sparsity issue [24, 30-33]. Such model-based systems use various clustering algorithms to create groups of users with similar tastes [20-25]. Although the clustering approach performs better than pure CF, these approaches lead to increased computational complexity. To address the issues of the Recommendation System, there is a need for a hybrid optimized algorithm that can overcome the flaws and present novel and personalized recommendations.

The primary objective of this research is to integrate an Elephas-Kiboko Optimization (EKO) into an integrated distance metric-based CNN model for implementing a movie recommendation system. This integration enhances the recommendation's accuracy by utilizing adaptive mechanisms. Using CNN can efficiently capture the hidden relations and enhance the accuracy of recommendations [34]. The Elephas-Kiboko optimization technique enhances the performance of algorithms by developing learning mechanisms that effectively balance the exploration and exploitation phases within the EK framework. The self-updating properties of the EK clan provide a diverse set of recommendations.

Furthermore, its memory-based escape mechanisms improve the system's resilience to challenges like cold start and data sparsity, leading to more accurate, personalized movie recommendations that adapt to users' changing preferences and ensure a more tailored experience. Also, this optimization enhances CNN's performance by refining its structure and parameters, resulting in better capturing of the diverse features of movie data. This refinement not only strengthens the classification function for more precise movie recommendations but also ensures optimal efficiency in handling dynamic content updates. The approach fosters a close alignment between classification, user preferences, and movie characteristics, providing consistently relevant, personalized recommendations as user tastes evolve.

The structure of the manuscript is organized as follows: Section 2 provides a review of recent studies, their methodologies, and associated challenges. Section 3 introduces the methodology for an efficient movie recommendation model. Section 4 presents the results obtained from applying the proposed model, and Section 5 concludes the manuscript.

2. Literature Review

Zahra Zamanzadeh Darbana and Mohammad Hadi Valipourb [1] proposed a recommendation algorithm based on a graph model. It combined apparent user rating similarities, demographic data, and users' location. This approach efficiently coped with the cold-start problem but did not work in real scenarios since all ratings had to be considered in the processing of the model.

Yongheng Mu and Yun Wu [2] suggested a multimodal movie recommendation system. This system combined movie and user features in an embedding vector fed to CNN for feature extraction and recommendation. This model was tested on a standard movie dataset and gave promising RMSE scores. However, the model failed to consider the dynamic and changing user preferences.

In [3], authors Yun Liu and Jun Miyazaki developed a Knowledge-Aware Attentional Neural Network (KANN) based movie recommendation system. This model utilized the entities extracted from movie reviews and considered the user-movie embedding. The model was tested and evaluated on the Imdb and Amazon datasets, and highly accurate recommendations were provided, but this system suffered from computational issues.

Authors Zhenlu Liang et al. presented a weight-normalized movie recommendation model (SCLW_MCRec) in [4]. This model was developed using a three-way neural interaction network. This model used one-dimensional CNN to capture the hidden embedding in user-item interactions. It efficiently addressed the cold-start problem and observed significant recommendation accuracy but lacked an automatic mechanism for selecting and constructing meta-paths based on data interactions.

Shadi Alzubi et al. [5] developed a TF-IDF-based approach, combined with cosine similarity, to analyze the movie features like cast, crew, keywords, and genres. This method noted better recommendation accuracy but required additional user features to improve the model's recommendation quality.

Sandipan Sahu et al. [6] introduced an early-stage recommendation model based on movie features like genre, cast, director, keywords, and description. This content based Movie Recommendation system then combined the user ratings with the primary features and fed it to CNN to predict movie recommendations. The proposed system achieved recommendation accuracy but suffered from issues like data sparsity.

Bharti Sharma et al. [7] proposed a movie recommendation system that utilized a bio-inspired Hybrid Sparrow-Clustering (HSC) algorithm to cluster the user and movie based on similar preferences. This model was implemented and tested on the Movie Lens dataset and demonstrated promising results in generating personalized recommendations.

In [8], authors CheonSol Lee et al. presented a new graph-based movie recommender system. This model combined sentiment, user emotion and user ratings to predict the missing user-item interactions. The model was optimized using BERT, and the use of IGMC provided promising and personalized recommendations.

R. Katarya [9] presented a movie recommendation system implementing the Artificial Bee Colony (ABC) algorithm. This meta-heuristic method relied on user ratings and user preferences to recommend movies. This optimization algorithm optimized the selection process of this user information for the generation of more personalized suggestions. The system was implemented on the Movie Lens dataset and demonstrated substantial improvement in recommendation quality.

The paper [10] proposed an attention mechanism-based recommendation system optimized using the Swarm Intelligence algorithm. It took into consideration the series of a user's interactions over a period of time. The model utilized the latest user-movie interactions to predict future interactions. The optimization algorithm identified the user behavior patterns and improved the accuracy of the recommendation.

Vellaichamy, Vimala, and Kalimuthu [11] proposed a collaborative filtering-based movie recommendation system optimised using the Bat Optimization algorithm. This system analysed the patterns of the user rating to form clusters of similar users and utilised the Bat optimization algorithm to finely tune the recommendations. The model was implemented on the Movielens dataset, effectively addressing challenges like data sparsity and cold start problems.

In another paper [12], authors by R Katarya and O P Verma presented a collaborative filtering-based movie recommender system optimized using the Cuckoo Search (CS) algorithm. The system considered the interaction between users and movies to generate recommendations. The use of the Cuckoo Search Optimization minimized the error in rating predictions. The experimental results depicted enhancement in prediction accuracy and computational efficiency.

The work presented by Senbagaraman, Senthilkumar, Subasankar, and Indira [13] was a hybrid movie recommendation system that combined model based collaborative filtering optimized using the Cuttlefish Optimization (CFO).

The Cuttlefish Optimization was applied to fine-tune the process of recommendation and minimize prediction errors of the user rating. The authors demonstrated that the combination of collaborative filtering and Cuttlefish Optimization improved the quality of recommendation accuracy and computational performance.

2.1. Motivation of Research

Although the combination of meta-heuristic algorithms and collaborative filtering enhances the performance of movie recommender systems, these methods are computationally expensive and suffer from scalability issues in real-time applications. Also, these systems suffer from cold start issues as the dataset has sparse user ratings. Therefore, there is a need to develop a meta-heuristic-based hybrid recommendation mechanism that can enhance the recommendation novelty and diversity.

2.2. Problem Statement

The Elephas Kiboko Optimized Integrated Distance enables the Convolutional Neural Network (EK-IDCNN) model, designed to generate movie recommendations that match the dynamically changing user's preferences by providing relevant content. It addresses key challenges and generates top-notch movie recommendations.

The proposed model processes the review data denoted as Ed and applies advanced feature extraction techniques and improved preprocessing steps. The EK-IDCNN model delivers effective recommendations by suggesting movies that are similar and relevant to the user, minimizing the error function and reducing the discrepancy between the model's output and the target data. The error function represented by Equation (1) focuses on enhancing the model's performance.

$$Error\ (fn) = (De + Db)/2 \tag{1}$$

Where D_b denotes the Bhattacharya distance, and D_e denotes the Euclidean distance.

3. Proposed Methodology

Movie recommendation systems have emerged as a crucial research domain owing to the significant use of the internet worldwide. The primary aim of this research is to provide remarkable improvement in the Movie Recommendation System. The first phase involves collecting movie review data from a standard dataset [26]. These movie reviews are users' textual feedback about the movie.

This data is then preprocessed by using various NLP methods like lemmatization, stop-word removal, and stemming so that it can be analyzed. This cleaned data is then used for feature extraction. Features like weighted rating, heterogeneous graph-based features, and features based on similarity metrics are extracted. These comprehensive features are then utilized by the proposed model to generate recommendations.

Using multiple features in the system leads to dimensionality reduction and improves computational efficiency and processing speed. Once the features are extracted, they are fed as input into the CNN model. This model uses the integrated distance metric, which is the combination of Bhattacharya and Euclidean distances, to calculate the similarity between movies. This step allows the system to determine how closely different movies are related based on their attributes.

Additionally, user preferences are inferred from the reviews, making it possible to recommend relevant movies by evaluating interactions between user ratings and items. Finally, the system implements the Elephas-Kiboko Optimization to fine-tune the CNN model and enhance the prediction accuracy. Figure 1 shows the entire flow of the proposed work.

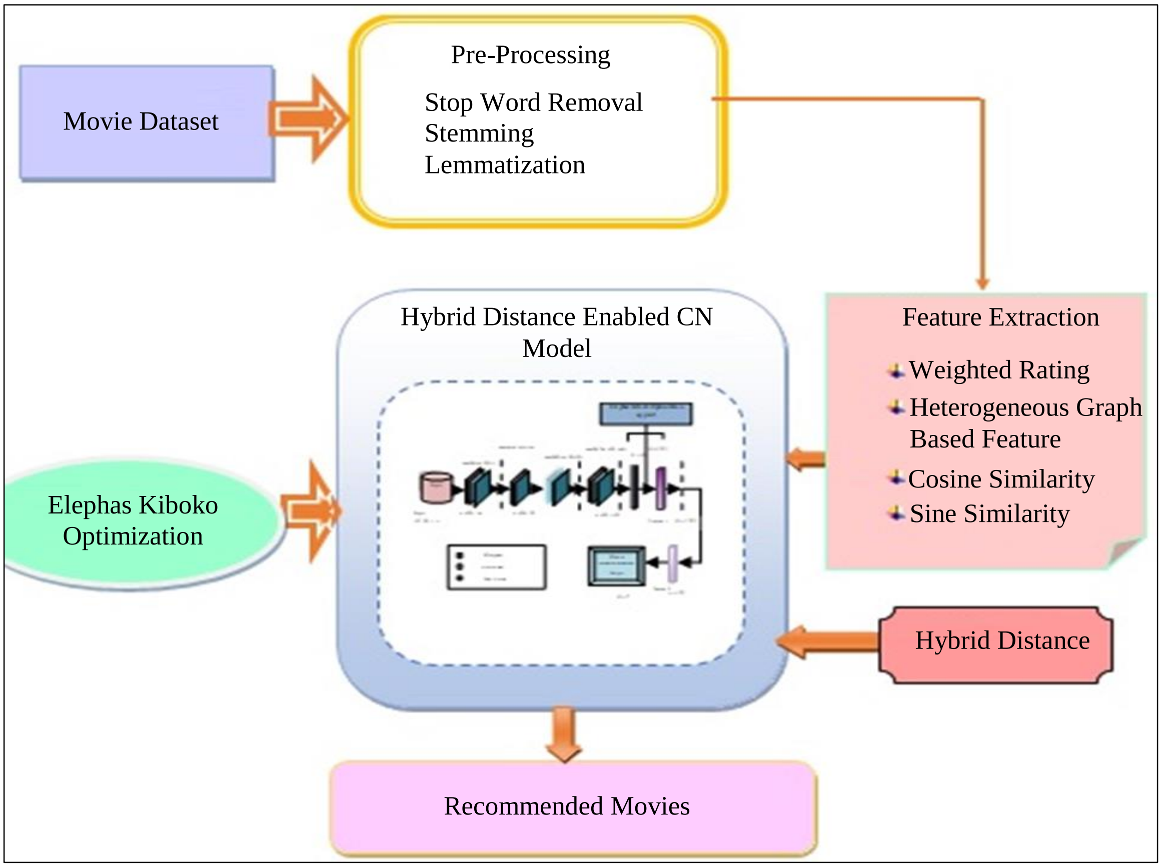


Fig. 1 The proposed system model

3.1. Input and Pre-Processing

The standard dataset tmdb [26] is used as the input to the proposed model and is represented as E_d. Data pre-processing is performed on the E_d, which involves several steps, such as lemmatization, stop word removal, and stemming. These pre-processing steps are designed to standardize and enhance the quality of text data, thereby making it simpler to analyze. The pre-processed output is denoted as Ed*.

3.2. Feature Extraction and Concatenation

The feature extraction step involves the calculation of weighted ratings, the use of Heterogeneous Graphs for movie features, and the finding of sine and cosine similarity among movies. The weighted rating ensures that the movies with higher ratings are prioritized in the recommendation process. Also, the incorporation of Heterogeneous Graphs (HG) to represent special features makes the model capable of capturing the complex relationships between users and movies. The sine and cosine similarities establish relations among similar movies, thereby efficiently suggesting the most relevant movies to users. The mathematical representation of these features over input dataset Ed* is shown in Equations (2), (3) and (4), respectively.

$$Weighted\ Rating\ WR(E_d^*) = \frac{c}{c+n}S + \frac{n}{c+n}D \tag{2}$$

Where, c is the number of votes for the movie, n is the minimum vote required to be listed in the chart, S is the average rating of the movie computed from the pre-processed data, and D is the mean vote across the entire dataset.

$$CS(P,R) = \sum_{j=1}^{n} \frac{P_j \times R_j}{\sqrt{\sum_j P_j^2 . \sqrt{\sum_j R_j^2}}} \tag{3}$$

$$SS(P,R) = \frac{P \times R}{\|P\|\ \|R\|} \tag{4}$$

Where, P and R and two movies with Pj and Rj Features.

Finally, these features are concatenated into Yi, as shown in Equation (5), so it can be fed as an input to the Integrated Distance Enabled Convolutional Neural Network (ID-CN).

$$Y_i = WR(Ed^*) + HG(Ed^*) + CS(Ed^*) + SS(Ed^*) \tag{5}$$

3.3. Elephas Kiboko Optimized Integrated Distance Enabled Convolutional Neural Network (EK-IDCNN)

The concatenated feature output Yi serves as the input for the Movie Recommendation Model. The proposed IDCNN uses various distance measures to improve the recommendation efficiency. This model utilizes the CNN to learn the hidden representations of user reviews from the input data. The integrated distance method combines Bhattacharyya distance and Euclidean distance to best capture the similarity between two feature vectors. The proposed CNN model employs conv2D layers to obtain hidden hierarchical features from input data. These layers apply filters over the input data, which enables the network to learn complex spatial relationships and dependencies.

This hierarchical approach is capable of representing features extracted from the movie reviews. To enhance the expressiveness of the model, non-linear activation functions are integrated into the CNN. Functions, like Rectified Linear Units (ReLU) are used to introduce non-linearity in the network and enable the network to learn and model the relationships between features extracted in earlier layers. This is especially useful when analyzing intricate details like sentiments in input data. Another crucial component of the CNN architecture is the pooling layer that subsamples the feature maps created by the conv2D layers. Its role is to reduce spatial resolution while retaining essential information. As a result, this abstraction process helps identify the most important features and optimizes computational efficiency.

In a Deep CNN, the fully connected layer processes the pooled features to analyze higher-level relationships and generate predictions. This integrates both spatial and semantic features from textual and visual inputs and enables the network to capture complex relationships essential for modelling user preferences and movie characteristics. The fully connected layer then uses the learned weights, biases, and activation functions like ReLU to produce the final predictions and generate accurate and personalized movie recommendations. The architecture of the proposed Convolution Network is shown in Figure 2. The goal of the proposed model is to minimize the error function described in Equation (1) by reducing the difference between the output and target pairs.

The Optimization Algorithm: The Elephas Kiboko Optimization Algorithm (EKA) enhances the working of a neural network by integrating global search strategies inspired by elephants and local optimization techniques based on hippopotamuses. These methods enable the network to identify more accurate feature representations, boost adaptability, and enhance performance in complex prediction tasks. The algorithm’s balance between exploration and adaptive refinement makes it a highly effective optimization tool for deep learning models. The flowchart of the Elephas Kiboko Optimization Algorithm is illustrated in Figure 3. EKA is a population-based optimization method that falls in the category of genetic algorithms or bio-inspired optimization algorithms. The fundamental concept is that multiple search agents work together to explore the search space, seeking optimal solutions by adjusting the values of decision variables. The optimization process at the initial step creates a population of randomly generated initial solutions. The algorithm assigns a random position X to each decision variable for each search agent. This position is determined by sampling a random value s, which adjusts the range between the variable's lower bound (lj) and upper bound (uj). The randomness ensures that the initial solutions are distributed across the search space. After the initial positions of the agents are determined, they begin exploring the search space, modifying their positions in order to find an optimal solution.

The goal of the optimization algorithm is to find the solution X_t that minimizes the error and ensures accuracy in the model's predictions. The next step is either to explore new resources to expand the user's horizon or to exploit them by recommending similar resources. The choice between exploration and exploitation depends on the relationship between the probability of encountering obstacles PE and the expected number of optimal region LE. If PE<LE, exploration for new resources is conducted, whereas if PE≥LE, exploitation is carried out. Exploration is the process of finding new possible relationships between the features among the search space. It can be conducted either as internal exploration or as external exploration.

Internal exploration refers to an iterative process of finding a stable position for Xt within the search space that exhibits features similar to those of the target user. On the other hand, external exploration refers to finding a similar feature map outside the search space and, if found, updating the exploration or returning to the best position of search. Finally, when the number of possible iterations is exhausted, the algorithm terminates and provides the optimal set of predictions.

4. Result and Discussion

4.1. Experimental Setup

The model is developed using Python script on a Windows 10 system with 8 GB RAM.

4.2. Dataset Used

Movie Lens datasets [21] are utilized to build the final recommendation system. The Complete Dataset includes 270,000 users, 750,000 tag applications, and 26 million ratings for 45,000 movies. It also incorporates tag genome data, consisting of 1,100 tags with 12 million relevance ratings. The Small Dataset contains 1,300 tag applications from 700 users, covering 9,000 movies and 100,000 ratings. The proposed system is developed using a publicly available static dataset. The research does not gather new user data, and therefore, there are no immediate ethical concerns regarding data collection, consent, or privacy violations in the context of this research.

4.3. Experimental Results

The experimental results are implemented based on two factors, i.e., K-fold and training percentage.

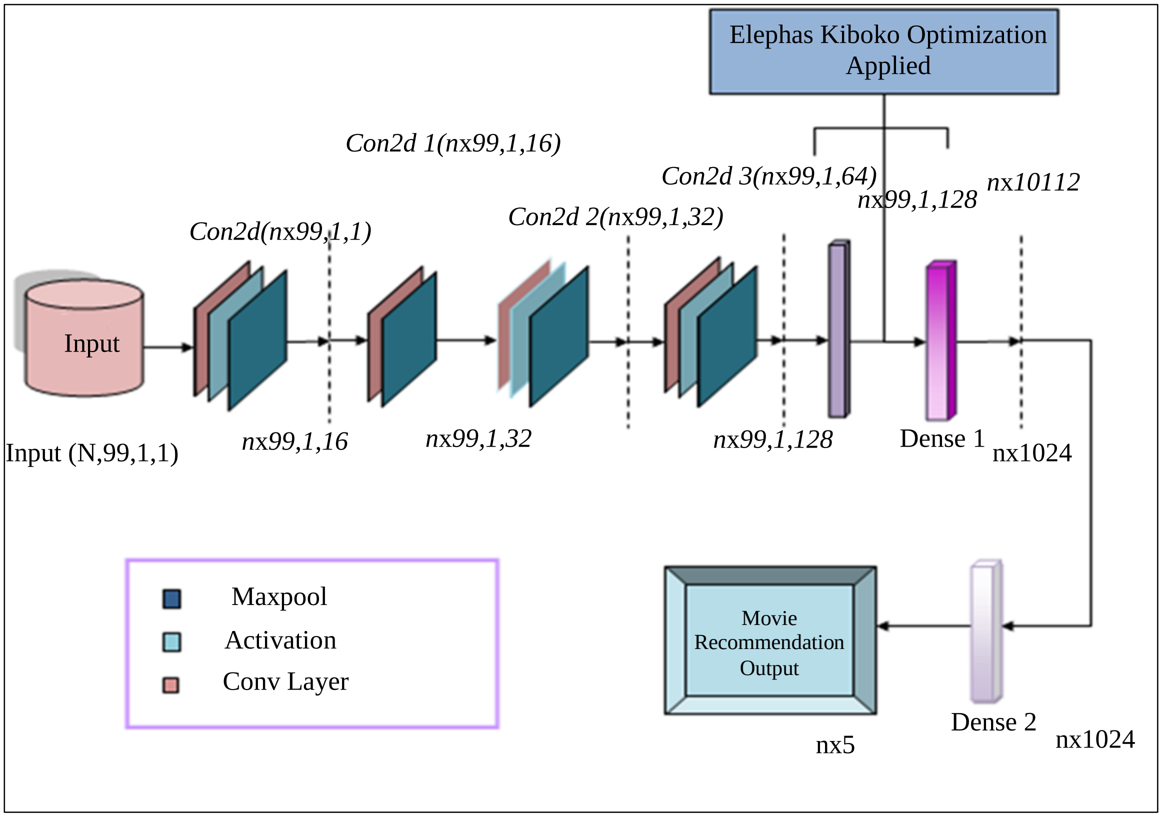


Fig. 2 Proposed CNN architecture

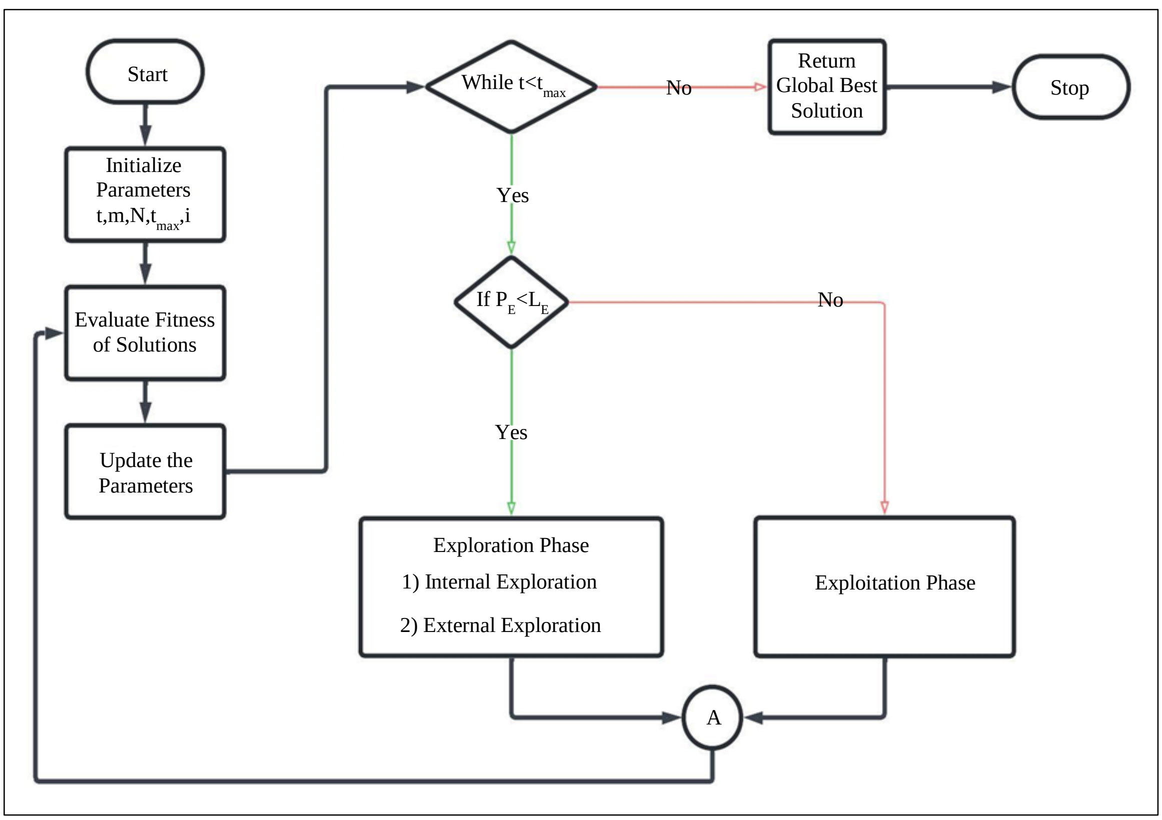


Fig. 3 Flowchart of Elephas-Kiboko optimization algorithm

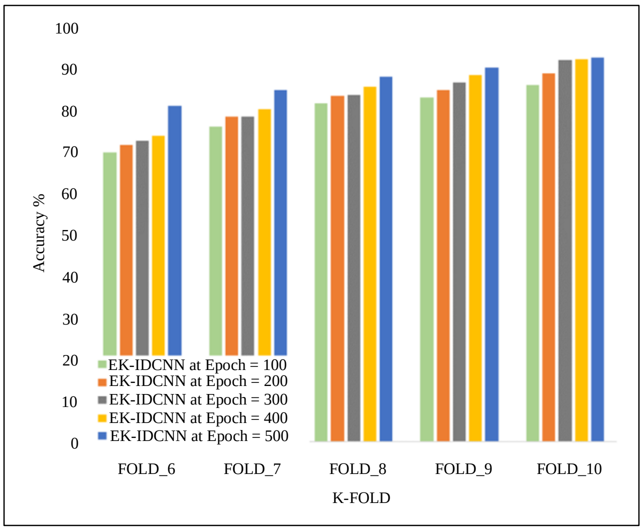
4.3.1. Performance Analysis on the Basis of K-Fold and Training Percentage

The results of the movie recommendation system developed using the EK-IDCNN model based on K-fold are presented in Figure 4. Figure 4(a) shows the accuracy percentages at epochs 100, 200, 300, 400, and 500, with a 10-fold cross-validation achieving 84.80%, 87.5%, 90.7 %, 91.0%, and 91.5%, respectively. Figure 4(b) displays the F1-score results for the EK-IDCNN model with a 10-fold cross-validation, yielding 83.60%, 85.01%, 87.5%, 90.0%, and 91.39% at the corresponding epochs. The precision results for the same epoch values are shown in Figure 4(c), with percentages of 86.27%, 87.25%, 89.92%, 91.25%, and 92.51% at a 10-fold cross-validation. Lastly, Figure 4(d) presents the recall percentages for the same epochs, which are 80.22%, 81.00%, 85.88%, 87.40%, and 90.29% at a 10-fold cross-validation.

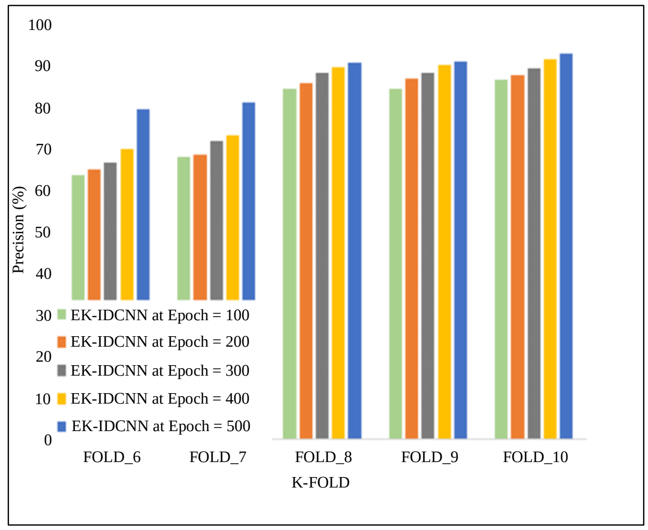
The results of the movie recommendation system developed using the EK-IDCNN model based on Training Percentage (TP) are shown in Figure 5.

Figure 5(a) displays the accuracy percentages at epochs 100, 200, 300, 400, and 500, with a TP of 80, achieving 86.2%, 86.3%, 87.9%, 90.13%, and 92.21%, respectively. Similarly, Figure 5(b) presents the F1-score results for the EK-IDCNN model at a TP of 80, which are 82.43%, 84.44%, 87.19%, 90.83%, and 93.12%.

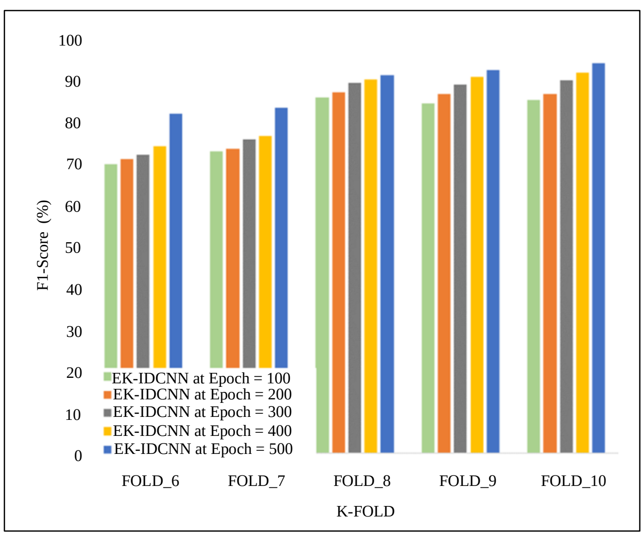
The precision results for the same epochs are shown in Figure 5(c), with percentages of 80.75%, 82.86%, 85.08%, 90.21%, and 91.47% at a TP of 80. Finally, Figure 5(d) displays the recall percentages for the same epoch values, which are 84.19%, 86.09%, 90.07%, 91.45%, and 95.54% at a TP of 80.



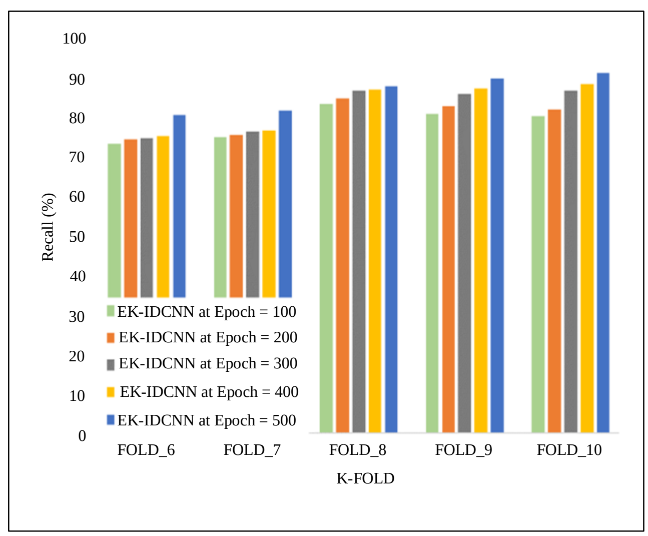
(a)



(c)



(b)



(d)

Fig. 4 Performance analysis based on K-fold a) Accuracy, b) F1-score, c) Precision, and d) Recall.

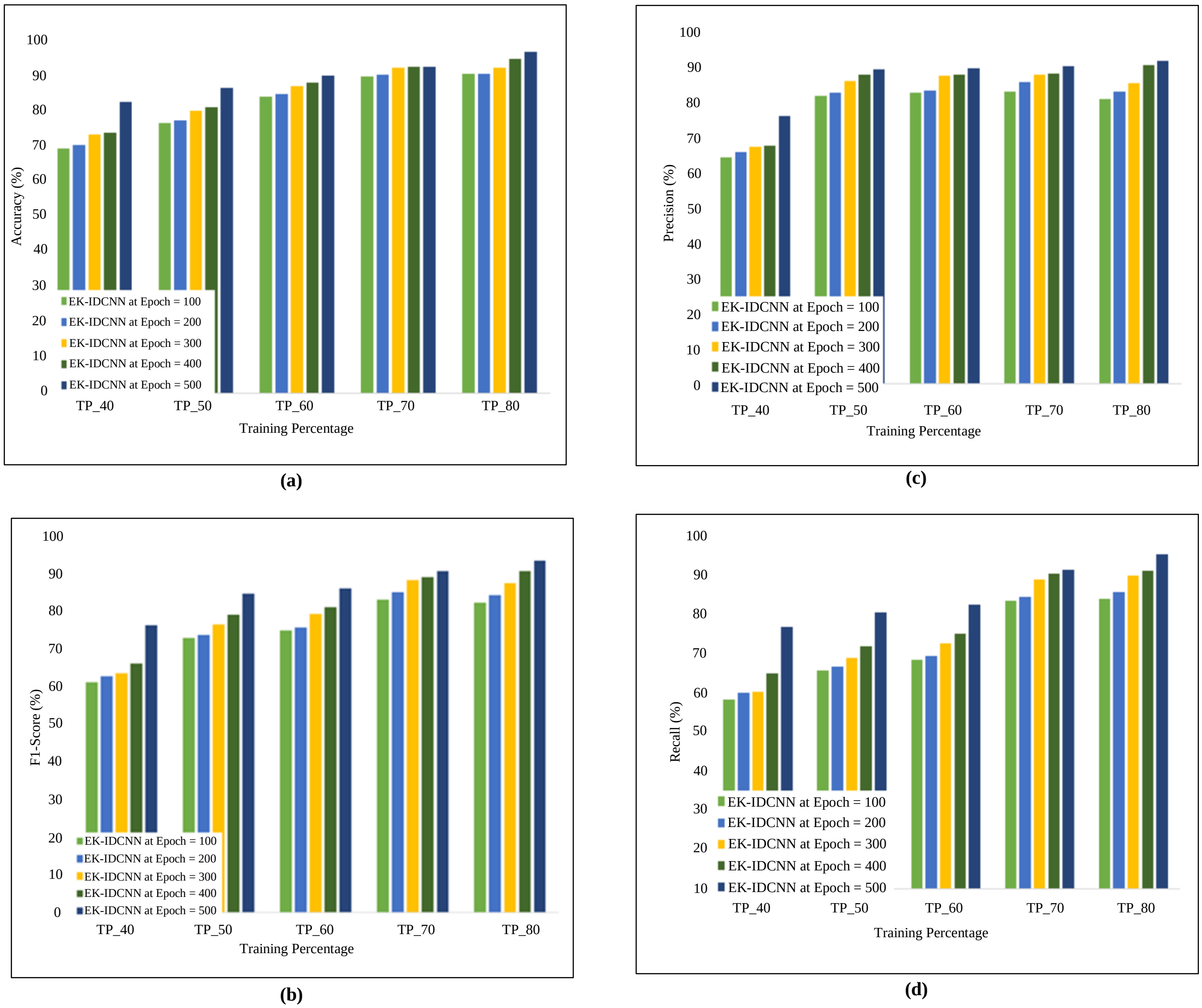


Fig. 5 Performance analysis based on TP a) Accuracy, b) F1-score, c) Precision, and d) Recall.

4.3.2. Comparative Analysis on the Basis of K-Fold and Training Percentage

Table 1 shows the comparison of the performance metric for various state-of-art models.

Comparison Based on K-Fold

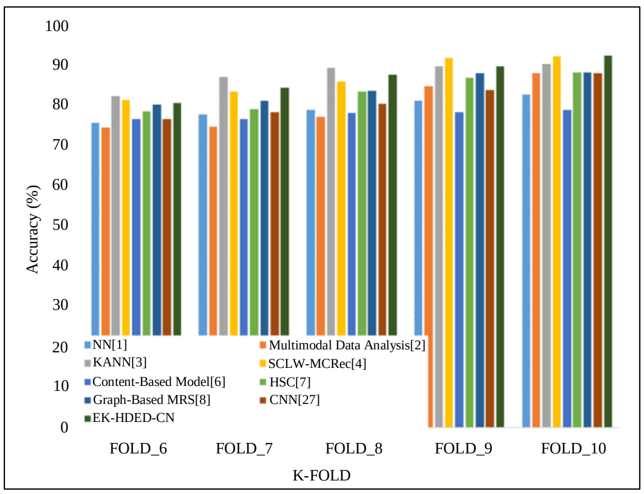
When comparing the proposed model with the other state-of-art models, it is observed that the EK-IDCNN model outperformed the graph-based movie recommender system, showing an improvement of 4.43% and achieving an accuracy of 93.43%, as shown in Figure 6(a). Compared to the graph-based movie recommender system in Figure 6(b), the EK-IDCNN model demonstrated a higher F1-score, surpassing it by 1.29% and achieving an F1-score of 92.5% with a 10-fold cross-validation. As illustrated in Figure 6(c), the EK-IDCNN model exhibited superior performance in the movie recommendation system, exceeding the graph-based system by 1.1%. It achieved a precision of 93.6% with a 10-fold cross-validation, outperforming the graph-based recommender. Finally, in Figure 6(d), the EK-IDCNN model achieved a higher recall than the graph-based movie recommender system, surpassing it by 2.27% and reaching a recall of 92.5% with a 10-fold cross-validation.

Comparison on the Basis of Training Percentage

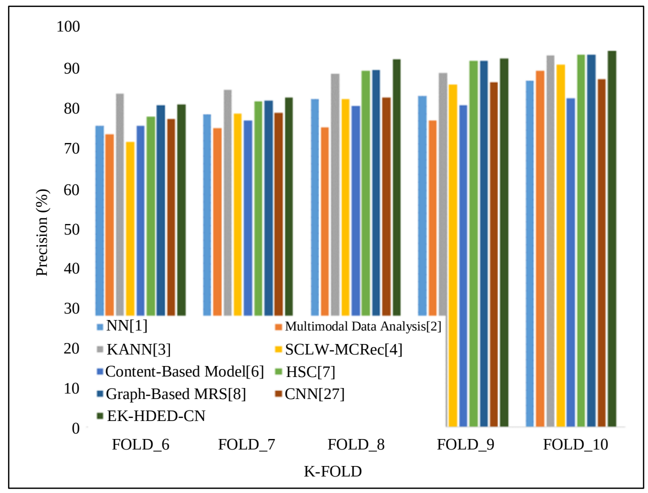
When comparing the proposed model with the other state-of-art models, it is observed that the EK-IDCNN model outperformed the graph-based movie recommender system, achieving a slight improvement of 1.71% in movie recommendation accuracy, reaching 94.87%, as shown in Figure 7(a). Compared to the graph-based system in Figure 7(b), the EK-IDCNN model achieved a higher F1-score, surpassing it by 0.4% with an F1-score of 94.65% at a TP of 90. As shown in Figure 7(c), the EK-IDCNN model demonstrated better performance in the movie recommendation system, outperforming the graph-based system by a small margin of 0.27%. It achieved a precision of 94.00% at a TP of 90, surpassing the graph-based recommender. Finally, in Figure 7(d), the EK-IDCNN model showed a higher recall than the graph-based movie recommender system, surpassing it by 0.25% and achieving a recall of 95.75% at a TP of 90.

Table 1. Comparative analysis of various state-of-art methods

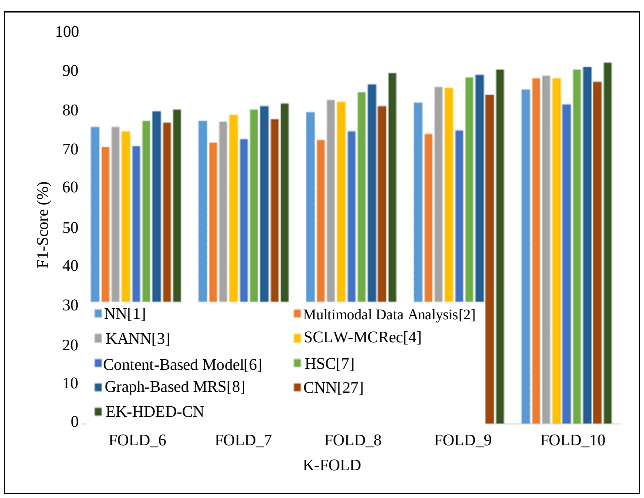
Models	TP 90				K-fold 10			
	Accuracy	F1-Score	Precision	Recall	Accuracy	F1-Score	Precision	Recall
Multimodal Data Analysis	92.68%	90.29%	87.5%	93.26%	88.25%	88.46%	88.54%	88.37%
KANN	93.4%	92.37%	90.01%	94.87%	90.48%	89.03%	92.40%	85.90%
SCLW-MCRec	88.87%	92.21%	90.76%	93.71%	92.39%	88.47%	90.01%	86.99%
Content-Based Filtering Model	93.11%	92.87%	92.53%	93.22%	79.14%	81.65%	81.59%	81.71%
NN	87.26%	91.16%	89.43%	92.95%	82.93%	85.58%	86.12%	85.05%
CN	89.47%	92.91%	91.7%	94.15%	88.18%	87.64%	86.37%	88.94%
HSC	91.11%	93.64%	92.68%	94.62%	88.44%	90.68%	92.44%	88.98%
Graph-Based MRS	93.16%	94.61%	93.74%	95.5%	88.48%	91.22%	92.54%	89.94%
EK-IDCNN	**94.87%**	**94.65%**	**94.00%**	**95.75%**	**93.43%**	**93.6%**	**92.20%**	**92.5%**



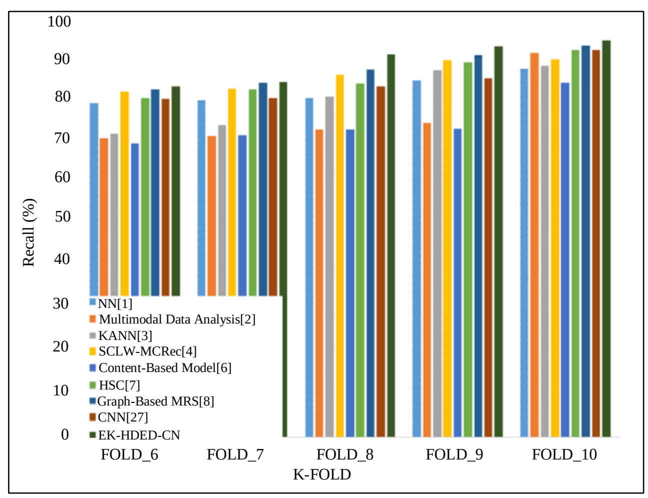
(a)



(c)



(b)



(d)

Fig. 6 Comparative analysis based on K-fold a) Accuracy, b) F1-score, c) Precision, and d) Recall.

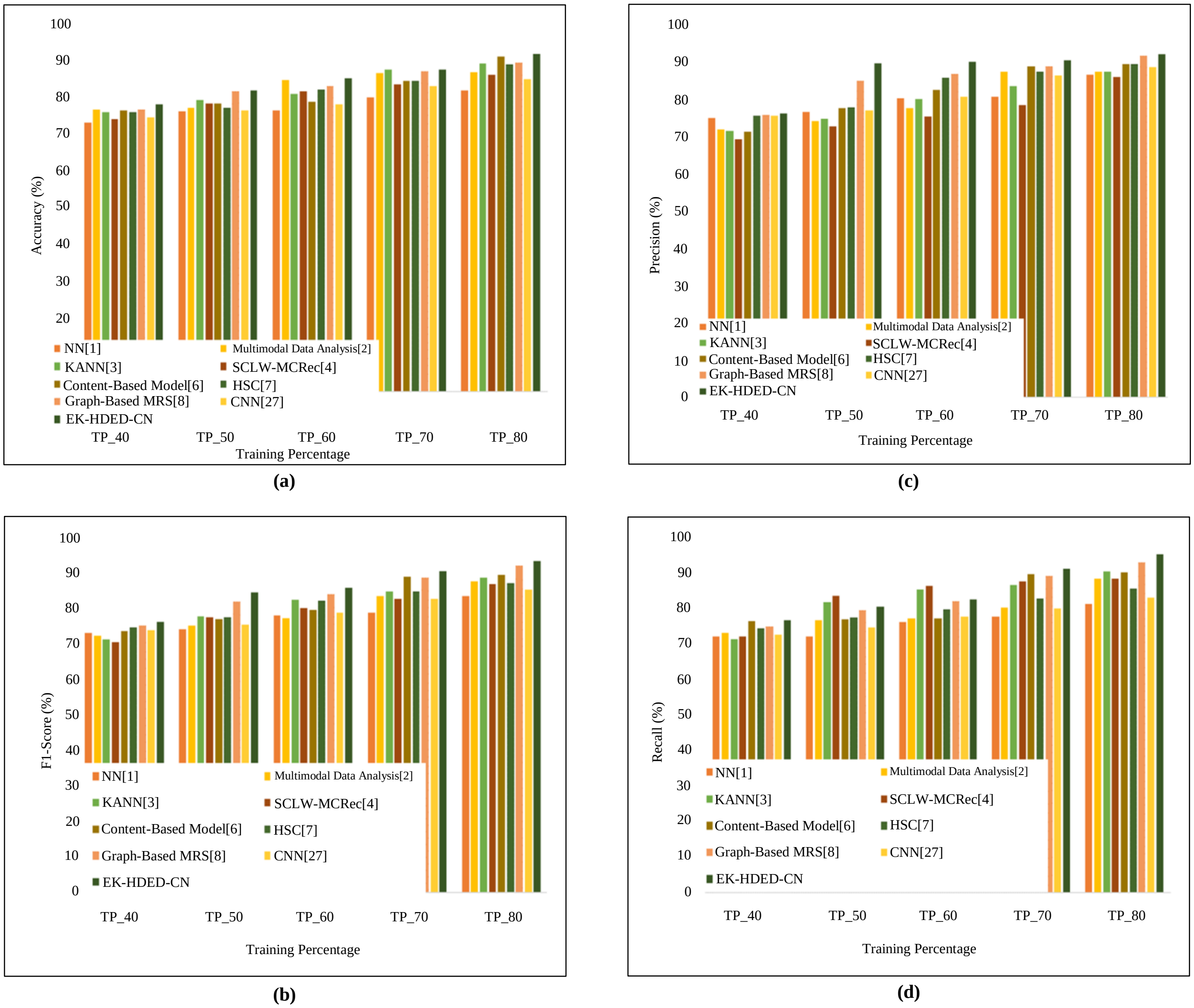


Fig. 7 Comparative analysis based on TP a) Accuracy, b) F1-score, c) Precision, and d) Recall.

4.4. Discussion

Existing movie recommendation models face numerous challenges like handling multimodal data, data sparsity, noisy datasets and cold-start problems. Multimodal data analysis techniques often struggle to integrate various data types seamlessly, while models like KANN and SCLW-MERec may encounter high computational costs and scalability issues. Content-based filtering models and traditional neural networks, such as NN and CN, frequently fail to capture complex user preferences and the interactions between users and items.

Hybrid Distance models, like HSC and graph-based recommenders, are often susceptible to overfitting and may not generalize well across diverse datasets. The proposed EK-IDCNN model addresses these challenges by incorporating the adaptive behaviors of the EK optimization, which enhances both exploration and exploitation capabilities. Integrating the hybrid distance metric with the CNN model ensures accuracy in the user-movie similarity evaluation. The use of the EK-optimization algorithm improves the personalization of recommendations. Using both textual and visual features handles cold-start issues, and combining multifaceted features, hybrid distance CNN and EK optimization provide robust and accurate recommendations. It also enhances the model's ability to handle large and sparse datasets.

5. Conclusion and Future Scope

This research presents the EK-IDCNN method, which leads to enhancement in the performance of a Movie Recommendation System. The proposed method addressed key challenges like Data Sparsity, dynamically changing user preferences and cold start problems. The system significantly improved the accuracy of recommendations and user satisfaction.

An Elephas-Kiboko optimization uses a dynamic strategy to optimize exploration and exploitation processes to improve the model's ability to identify and evaluate relevant features in the processed movie data using the hybrid distance metric. Therefore, integrating optimization and CNN techniques presented here represents a notable enhancement in creating effective and efficient recommendation systems.

The results demonstrate that the model performs exceptionally well with the movie recommender database, achieving high accuracy, F1 score, precision, and recall scores. Specifically, the results for TP 90 reach 94.87%, 94.65%, 94.00%, and 95.75%, respectively. The integration of integrated distance and bio-inspired algorithms has the advantage of exploration of diverse solutions. Also, fine-tuning and optimization provide diverse and optimal solutions even in large and sparse search spaces. These integrations provide efficient results in handling sparse data. However, the system may result in biased recommendations if the dataset is sufficiently diverse.

Future work could include the expansion of the dataset, which incorporates a broader range of user interactions and movie features, which could further improve the model's generalization capabilities. Also, integrating real-time data and user feedback could improve the system's flexibility. Moreover, the fusion of an advanced attention mechanism into the integrated model could further enhance recommendation accuracy.

Acknowledgments
The authors are thankful to the G.H. Raisoni University Amravati for providing research support to complete this work smoothly.

References
[1] Zahra Zamanzadeh Darban, and Mohammad Hadi Valipour, "GHRS: Graph-Based Hybrid Recommendation System with Application to Movie Recommendation," *Expert Systems with Applications*, vol. 200, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yongheng Mu, and Yun Wu, "Multimodal Movie Recommendation System Using Deep Learning," *Mathematics*, vol. 11, no. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yun Liu, and Jun Miyazaki, "Knowledge-Aware Attentional Neural Network for Review-Based Movie Recommendation with Explanations," *Neural Computing and Applications*, vol. 35, no. 3, pp. 2717-2735, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Zhenlu Liang, Zhisheng Yang, and Jingyong Cheng, "Weight Normalization Optimization Movie Recommendation Algorithm Based on Three-Way Neural Interaction Networks," *Complex & Intelligent Systems*, vol. 9, no. 6, pp. 6241-6254, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shadi AlZu'bi, Amjed Zraiqat, and Samar Hendawi, "Sustainable Development: A Semantics-Aware Trends for Movies Recommendation System using Modern NLP," *International Journal of Advances in Soft Computing & Its Applications*, vol. 14, no. 3, pp. 153-173, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Sandipan Sahu et al., "Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System," *IEEE Access*, vol. 10, pp. 42044-42060, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bharti Sharma et al., "Hybrid Sparrow Clustered (HSC) Algorithm for Top-N Recommendation System," *Symmetry*, vol. 14, no. 4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] CheonSol Lee et al., "Improving Graph-Based Movie Recommender System Using Cinematic Experience," *Applied Sciences*, vol. 12, no. 3, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Rahul Katarya et al., "Movie Recommender System with Metaheuristic Artificial Bee," *Neural Computing and Applications*, vol. 30, pp. 1983-1990, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Shuang Geng et al., "Swarm Enhanced Attentive Mechanism for Sequential Recommendation," *Advances in Swarm Intelligence*, pp. 442-453, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Vimala Vellaichamy, and Vivekanandan Kalimuthu, "Hybrid Collaborative Movie Recommender System Using Clustering and Bat Optimization," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 5, pp. 38-47, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Rahul Katarya, and Om Prakash Verma, "An Effective Collaborative Movie Recommender System with Cuckoo Search," *Egyptian Informatics Journal*, vol. 18, no. 2, pp. 105-112, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Senbagaraman M. et al., "A Movie Recommendation System Using Collaborative Approach and Cuttlefish Optimization," *International Conference on Emerging trends in Engineering, Science and Sustainable Technology*, pp. 95-99, 2017. [Google Scholar] [Publisher Link]

[14] Waleed Hattawi et al., "Recent Quality Models in BigData Applications," *International Conference on Information Technology*, Amman, Jordan, pp. 811-815, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mahesh Goyani, and Neha Chaurasiya, "A Review of Movie Recommendation System: Limitations, Survey and Challenges," *Electronic Letters on Computer Vision and Image Analysis*, vol. 19, no. 3, pp. 18-37, 2020. [Google Scholar] [Publisher Link]

[16] Ala Mughaid et al., "An Intelligent Cybersecurity System for Detecting Fake News in Social Media Websites," *Soft Computing*, vol. 26, no. 12, pp. 5577-5591, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] J. Bobadilla et al., "Recommender Systems Survey," *Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[18] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[19] Aghiles Salah, Nicoleta Rogovschi, and Mohamed Nadif, "A Dynamic Collaborative Filtering System via a Weighted Clustering Approach," *Neurocomputing*, vol. 175, pp. 206-215, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[20] Nikolaos Polatidis, and Christos K. Georgiadis, "A Multi-Level Collaborative Filtering Method that Improves Recommendations," *Expert Systems with Applications*, vol. 48, pp. 100-110, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Yehuda Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 426-434, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[22] Michael T. Lash, and Kang Zhao, "Early Predictions of Movie Success: The Who, What, and When of Protability," *Journal of Management Information Systems*, vol. 33, no. 3, pp. 874-903, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[23] Usman Ahmed, Humaira Waqas, and Muhammad Tanvir Afzal, "Pre-Production Box-Office Success Quotient Forecasting," *Soft Computing*, vol. 24, no. 9, pp. 6635-6653, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Nuning Kurniasih et al., "Online Media as a Movie Reference," *Journal of Physics: Conference Series*, vol. 1114, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shuai Zhang et al., "Deep Learning Based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1-38, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Rounak Banik, Movie Recommender Systems, Kaggle. [Online]. Available: https://www.kaggle.com/code/rounakbanik/movie-recommender-systems/input

[27] Haitao He et al., "Movie Recommendation System Based on Traditional Recommendation Algorithm and CNN Model," *Highlights in Science, Engineering and Technology*, vol. 34, pp. 255-261, 2023. [CrossRef] [Google Scholar] [Publisher Links]

[28] Pinar Civicioglu, and Erkan Besdok, "A Conceptual Comparison of the Cuckoo-Search, Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony Algorithms," *Artificial Intelligence Review*, vol. 39, pp. 315-346, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[29] Adil Fahad et al., "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267-279, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[30] Jon Herlocker, Joseph A. Konstan, and John Riedl, "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms," *Information Retrieval*, vol. 5, pp. 287-310, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[31] Joseph A. Konstan, and John Riedl, "Recommender Systems: from Algorithms to User Experience," *User Modeling and User-Adapted Interaction*, vol. 22, pp. 101-123, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[32] Xin Zhang, Jiwei Qin, and Jiong Zheng, "A Social Recommendation Based on Metric Learning and Users' Co-Occurrence Pattern," *Symmetry*, vol. 13, no. 11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[33] Ssvr Kumar Addagarla, and Anthoniraj Amalanathan, "Probabilistic Unsupervised Machine Learning Approach for a Similar Image Recommender System for E-Commerce," *Symmetry*, vol. 12, no. 11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[34] Swati Aswale et al., "Indian Classical Music Recognition Using Deep Convolution Neural Network," *International Journal of Electrical and Electronics Research*, vol. 12, no. 1, pp. 73-82, 2024. [CrossRef] [Google Scholar] [Publisher Link]